Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (previously presented):

A device comprising:

a port to receive one or more data streams, each data stream including one or more data frames;

a task scheduler coupled to the port, the task scheduler to generate a task identifier for every data frame received;

a first queue to hold task identifiers for which a corresponding data frame is of a first priority type;

a second queue to hold task identifiers for which the corresponding data frame is of a second priority type, the second priority type different than the first priority type;

a switch coupled to the first and second queues, the switch configured to retrieve task identifiers from the first queue and the second queue in a fair manner;

a third queue coupled to the switch, the third queue to hold a plurality of task identifiers placed in the third queue by the switch and provide the task identifiers to a processing unit in the order task identifiers were placed in the third queue by the switch;

a classifier coupled to the port and to the first queue, the classifier to assign one of a plurality of priority types to every data frame received, the plurality of priority types including the first priority type and the second priority type, the classifier to monitor the first queue for an overflow condition and, if an overflow condition is detected, to reassign data frame priority types from the first priority type to the second priority type to prevent overflow of the first queue; and

a task router coupled to the task scheduler, the classifier, the first queue, and the second queue, the task router configured to

receive the task identifier from the task scheduler, the task identifier corresponding to a received data frame,

receive a priority type from the classifier, the priority type corresponding to the received data frame,

place the task identifier in the first queue if the priority type is the first priority type, and

place the task identifier in the second queue if the priority type is the second priority type.

2. (cancelled)

3. (previously presented):

The device of claim 1 further comprising:

a look-up table communicatively coupled to the task scheduler and to the port, the look-up table to provide one of the first priority type and the second priority type to the task scheduler for every data frame received according to the data stream in which the data frame was included.

4. (previously presented):

The device of claim 3 wherein one of the first priority type and the second priority type is pre-assigned to the data stream.

5. (previously presented):

The device of claim 3 wherein the conversions between priority types and data frame types are dynamically configured in response to usage of the first and second queues.

6.-7. (cancelled)

8. (previously presented):

The device of claim 1 wherein the switch is configured to retrieve task identifiers from both the first and second queues in a task retrieval cycle in which at least one task identifier is retrieved from each of the first and second queues such that space in the third queue is allotted equally according to processing time restrictions.

9. (previously presented):

The device of claim 1 wherein the switch is configured to retrieve task identifiers with the first priority type until a cumulative processing time requirement for the retrieved task identifiers with the first priority type is substantially equal to a processing time requirement for the task identifiers with the second priority type, and then to retrieve a task identifier with the second priority type.

10. (original):

The device of claim 1 wherein the third queue is a shared execution queue from which one or more processing units retrieve task identifiers to process.

11. (currently amended):

A method comprising:

receiving one or more data streams, each data stream including one or more data frames of one or more data frame ~~types;~~ types, each data frame type corresponding to a particular processing time requirement for data frames of the data frame type;

determining a task priority level for each data frame received;

routing each data frame to one of one or more storage queues based on the task priority level of each data frame;

retrieving the data frames from the one or more storage queues during a task retrieval cycle according to a fair and weighted processing scheme based on task priority ~~level;~~ ~~and~~ level, wherein data frames of approximately equal total processing time restrictions are retrieved from each storage queue in a task retrieval cycle;

reassigning the task priority level for each data frame received prior to routing if an overflow condition is detected in a first storage queue and if the task priority level would cause a data frame to be stored in the first storage queue, the task priority level being reassigned to a task priority level that will cause the data frame to be stored in other than the first storage ~~queue.~~ queue; and

placing a plurality of the retrieved data frames into an execution queue to be processed by a processing unit.

12. (previously presented):

The method of claim 11 wherein the task priority level is determined from one of frame size, echo canceller tail length, codec type, and frame processing requirements.

13. (original):

The method of claim 11 wherein the task priority level corresponding to a particular data frame type is pre-configured.

14. (original):

The method of claim 11 wherein each storage queue stores data frames of a different task priority level than the other storage queues.

15.-18. (cancelled)

19. (currently amended):

A method comprising:

receiving one or more data streams, each data stream including one or more data frames of one or more data frame ~~types;~~ types, each data frame type corresponding to a particular processing time requirement for data frames of the data frame type;

determining a task priority level for each data frame received;

assigning a unique task identifier to each received data frame;

storing each task identifier to one of multiple storage queues according to the task priority level of the corresponding data frame;

retrieving task identifiers from the one or more storage queues during a task retrieval cycle according to a weighted processing scheme based on task priority ~~levels; and~~ levels, wherein task identifiers corresponding to data frames of approximately equal total processing time requirements are retrieved from each storage queue in a task retrieval cycle;

reassigning the task priority level for each data frame received prior to storing each task identifier if an overflow condition is detected in a first storage queue and if the task priority level would cause a task identifier to be stored in a first storage queue, the task priority level being reassigned to a task priority level that will cause the task identifier to be stored in other than the first storage ~~queue. queue; and~~

placing a plurality of the retrieved task identifiers into an execution queue to be processed by a processing unit.

20. (previously presented):

The method of claim 19 wherein the task priority level is determined from one of frame size, echo canceller tail length, codec type, and frame processing requirements.

21.-23. (cancelled):

24. (currently amended):

A machine-readable medium having one or more instructions for scheduling processing tasks, which when executed by a processor, causes the processor to perform operations comprising:

receiving one or more data streams, each data stream including one or more data frames of one or more data frame ~~types;~~ types, each data frame type corresponding to a particular processing time requirement for data frames of the data frame type;

determining the task priority level for each data frame received;

routing each data frame to one of one or more storage queues based on the task priority level of each data frame; and

retrieving the data frames from the one or more storage queues during a task retrieval cycle according to a fair and weighted processing scheme based on task priority ~~level;~~ ~~and~~ level, wherein data frames of approximately equal total processing time are retrieved from each storage queue in a task retrieval cycle;

reassigning the task priority level for each data frame received prior to routing if an overflow condition is detected in a first storage queue and if the task priority level would cause a data frame to be stored in the first storage queue, the task priority level being reassigned to a task priority level that will cause the data frame to be stored in other than the first storage ~~queue.~~ queue; and

placing a plurality of the retrieved data frames into an execution queue to be processed by a processing unit.

25. (previously presented):

The machine-readable medium of claim 24 wherein the task priority level is determined from one of frame size, echo canceller tail length, codec type, and frame processing requirements.

26. (original):

The machine-readable medium of claim 24 wherein each storage queue stores data frames of a different task priority level than the other storage queues.

27.-30. (cancelled)

Conclusion

Applicant reserves all rights with respect to the applicability of the doctrine of equivalents. Applicant respectfully requests that a timely Notice of Allowance be issued in this case.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Dated: July 18, 2006

By _____

James Henry
Reg. No. 41,064
Tel.: (714) 557-3800 (Pacific Coast)